**Logo

Description automatically generated**

**San Francisco Bay University**

**Programming in C Language**

**Homework Assignment #1**

**Due day: 6/3/2023**

**Instruction:**

1. **Push the source code to Github or piazza platform.**
2. **Please follow the code style rule like programs on handout.**
3. **Overdue homework submission could not be accepted.**

**4. Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

1. Write a program to read-in numbers (integer) from keyboard and save them to an array, and then sort them ascendingly by bubble sorting method. After that, print them out.

*Output*

*Enter a number of array’s size for a series of numbers saving: 5*

*Enter a series of numbers: 5 6 2 7 1*

*After sorting, output sequence: 1 2 5 6 7*

#include <stdio.h>

void buble\_sort(int arr[], int n){

    for (int i=0; i<n-1; i++){

        for (int j=0; j<n-i-1; j++){

            if (arr[j]>arr[j+1]){

                int temp = arr[j];

                arr[j] = arr[j+1];

                arr[j+1] = temp;

            }

        }

    }

}

int main(){

    int array\_size;

    printf("Enter a number of array size for a series of numbers saving: ");

    scanf("%d", &array\_size);

    int input\_num[array\_size];

    printf("Enter a series of numbers: ");

    for (int j =0; j<array\_size; j++){

        scanf("%d", &input\_num[j]);

    }

    buble\_sort(input\_num, array\_size);

    printf("After sorting, output sequence: ");

    for (int j =0; j<array\_size; j++){

        printf("%d ", input\_num[j]);

    }

    return 0;

}

2. It is similar to the above question, and just print ascendingly sorted odd numbers first and followed by sorted even numbers.

*Output*

*Enter a number of array’s size for a series of numbers saving: 5*

*Enter a series of numbers: 5 6 2 7 1*

*After sorting, output sequence: 1 5 7 2 6*

*#include* <stdio.h>  
*// question 1  
void* buble\_sort(*int* arr[], *int* n){  
 *for* (*int* i=0; i<n-1; i++){  
 *for* (*int* j=0; j<n-i-1; j++){  
 *if* (arr[j]>arr[j+1]){  
 *int* temp = arr[j];  
 arr[j] = arr[j+1];  
 arr[j+1] = temp;  
 }  
 }  
 }  
 *// printing the odd numbers  
 for* (*int* i=0; i<n; i++){  
 *if* (arr[i]%2 == 1){  
 printf("%d ", arr[i]);  
 }  
 }  
  
 *// printing the even numbers  
 for* (*int* i=0; i<n; i++){  
 *if* (arr[i]%2 == 0){  
 printf("%d ", arr[i]);  
 }  
 }  
  
}  
  
*int* main(){  
 *int* array\_size;  
 *int* arr[array\_size];  
 printf("Enter a number of array size for a series of numbers saving: ");  
 scanf("%d", &array\_size);  
 printf("Enter a series of numbers: ");  
 *for* (*int* i=0; i<array\_size; i++){  
 scanf("%d", &arr[i]);  
 }  
 buble\_sort(arr, array\_size);  
  
 *return* 0;  
}

3. If there is a pancake and taking one cut, it will become 2 pieces. Of course, taking two cuts will create 4 pieces. Write a program to calculate how many pieces can be gotten if taking n cuts.

*Output*

*Enter how many cuts you want: 2*

*Pieces will be: 4*

*Output*

*Enter how many cuts you want: 3*

*Pieces will be: 7*

*#include* <stdio.h>  
  
*int* calculatePieces(*int* n) {  
 *if* (n == 0) {  
 *return* 1;  
 }  
 *return* 2 \* calculatePieces(n - 1);  
}  
  
*int* main() {  
 *int* n;  
 printf("Enter how many cuts you want: ");  
 scanf("%d", &n);  
  
 *int* numPieces = calculatePieces(n);  
 printf("Pieces will be: %d\n", numPieces);  
  
 *return* 0;  
}

4. Write a program to calculate how many *1*s is for a decimal number as input arguments

*Output*

*Enter decimal number: 15*

*There are 4 ones in given decimal number*

Notice that *15*’s binary number is *1111*

*#include* <stdio.h>  
*void* one\_counter(*int* n){  
 *int* count = 0;  
 *while* (n>0){  
 *if* (n%2 == 1){  
 count= count+1;  
 }  
 n = n/2;  
 }  
 printf("There are %d ones in given decimal number", count);  
}  
*int* main(){  
 *int* n;  
 printf("Enter decimal number: ");  
 scanf("%d", &n);  
 one\_counter(n);  
 *return* 0;  
}

5. There is a kind of bacterium with two sub-species A and B. They are very similar and difficult to differentiate, but the major difference between them is capability of reproduction. And reproduction in A is much stronger than that of B. Assuming that in a research center, researcher massed up Petri dishes with A and B bacterium, write a program to find which one is A, and which one is B in terms of each reproduction rate, given that reproduction rate is the ratio of new number of bacteria to original number after one hour (PR = new bacterial number / original bacterial numbers). Because of the huge different reproduction rate, it means that the difference of PR in any two Petri dishes belonging to the same sub-species is extremely smaller than that in any two Petri dishes belonging to the different sub-species.

*Output*

*Enter total number of Petri dishes: 5*

*Enter Petri dish label, original bacterial number, new bacterial number*

*after one hour reproduction:*

*1 10 3456*

*2 10 5644*

*3 10 4566*

*4 20 234*

*5 20 232*

*Running results:*

*3 in A* sub-species *and Petri dish labels from smaller PR to bigger PR are 1 3 2*

*2 in B* sub-species *and Petri dish labels from smaller PR to bigger PR are 5 4*